

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2002-358280  
(43)Date of publication of application : 13.12.2002

(51)Int.Cl. 606F 15/00

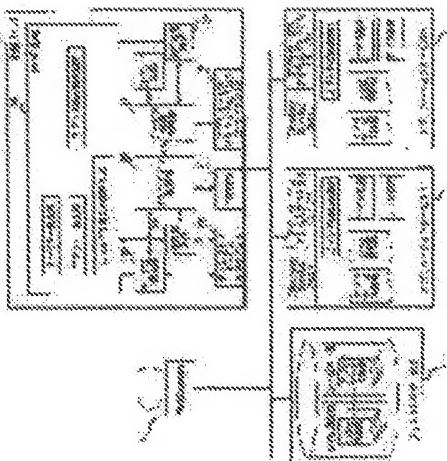
(21)Application number : 2001-166207 (71)Applicant : RICOH CO LTD  
(32)Date of filing : 31.05.2001 (72)Inventor : IMAI TATSUYA

## (84) CLIENT SERVER SYSTEM

### (57)Abstract:

**PROBLEM TO BE SOLVED:** To make it possible to use a new application service from a client terminal device without adding a new module to a relay server when the new application service is added on a network.

**SOLUTION:** In a client server system wherein the client terminal device 1, an application service device 2 having an application service means and the relay server 3 are connected through the network, and the relay server 3, depending on a request from the client terminal device 1, makes a request of the application service means for processing, and receives the processing result from the application service means to send it back to the client terminal device 1, the system is constituted such that the relay server 3 acquires the information on a processing request method regarding the application service designated from the client terminal device 1, and creates HTML data based on the information to send the data back to the client terminal device 1.



## LEGAL STATUS

- [Date of request for examination]
- [Date of sending the examiner's decision of rejection]
- [Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]
- [Date of final disposal for application]
- [Patent number]
- [Date of registration]
- [Number of appeal against examiner's decision of rejection]
- [Date of requesting appeal against examiner's decision of rejection]
- [Date of extinction of right]

(10)日本国特許庁 (JP)

## ○○ 公開特許公報 (A)

(11)特許出願公開番号

特開2002-358280

(P2002-358280A)

(12)公開日 平成14年12月12日 (2002. 12. 12)

601IntCl<sup>5</sup> 請求記載号  
G 06 F 15/00 310

F 1  
G 06 F 15/00 3103 60006

### 審査請求 水被求 請求項の数10 CL (全 13 回)

(20)出願番号 特願2001-165207(P2001-165207)

(71)出願人 009990767

株式会社リコー

東京都大田区中馬込1丁目3番6号

(22)出願日 平成13年5月31日 (2001. 5. 31)

(72)発明者 今井 雄也

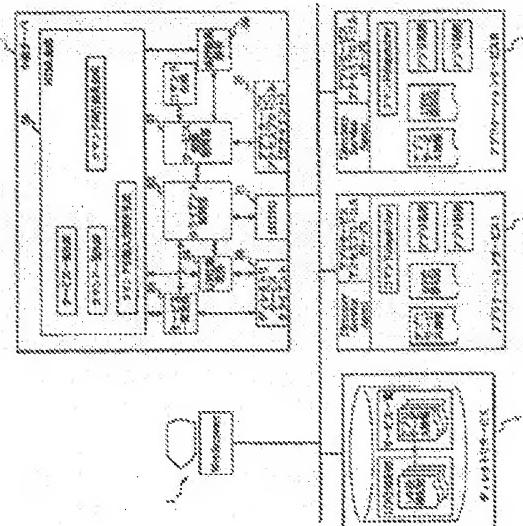
東京都大田区中馬込1丁目3番6号 株式  
会社リコー内

ドーム(郵便番号) 200-0013

### ○○ 【説明の各項】 クライアントサーバシステム

#### (57)【要約】

【課題】 ネットワーク上に新たなアプリケーションサービスが追加されたとき、中継サーバに新たなモジュールを追加することなくクライアント端末装置から新しいアプリケーションサービスを利用できるようにする。



【解決手段】 クライアント端末装置1、アプリケーションサービス手段を備えたアプリケーションサービス装置2、中継サーバ3がネットワークを介して接続され、その中継サーバ3がクライアント端末装置1からの要求によりアプリケーションサービス手段に処理を依頼し、処理結果をアプリケーションサービス手段から受信してクライアント端末装置1へ送達するクライアントサーバシステムにおいて、中継サーバ3が、クライアント端末装置1から指定されたアプリケーションサービスについて処理依頼方法に関する情報を取得し、その情報に基づいてBINデータを生成し、クライアント端末装置1へ返送する構成にした。

## 【特許請求の範囲】

【請求項1】 ブラウザを搭載したクライアント端末装置と、そのクライアント端末装置からの要求により処理を行う一つ以上のアプリケーションサービス手段と、HTTPサーバを含む中継サーバとがネットワークを介して接続され、その中継サーバがクライアント端末装置からの要求情報を受信し、その要求情報に基づいてアプリケーションサービス手段に処理を依頼し、処理結果をそのアプリケーションサービス手段から受信してクライアント端末装置へ返送するクライアントサーバシステムにおいて、中継サーバが、クライアント端末装置からの要求情報中に指定されたアプリケーションサービスについてサービスに対する処理依頼方法に関する情報を取得し、取得した情報を基づいてHTTPデータを生成し、クライアント端末装置へ返送することを特徴とするクライアントサーバシステム。

【請求項2】 請求項1記載のクライアントサーバシステムにおいて、クライアント端末装置で前記要求情報を作成するために、中継サーバの生成するHTTPデータが各アプリケーションサービスの各コマンドに対応した入力項目を備え、アプリケーションサービスがクライアント端末装置からの要求により複数の異なる処理を実行可能で、クライアント端末装置からの処理要求が処理内容に対応したコマンドを送信する形で往われる場合、前記入力項目を用いてクライアント端末装置のブラウザ上で入力または選択された内容を中継サーバへ送ることを特徴とするクライアントサーバシステム。

【請求項3】 請求項2記載のクライアントサーバシステムにおいて、アプリケーションサービス手段が各コマンドの実行にコマンドの指定の他にパラメータの指定が必要な場合、中継サーバが、前記ブラウザ上におけるパラメータ値の指定を可能な限りの入力項目をもつHTTPデータを生成することを特徴とするクライアントサーバシステム。

【請求項4】 請求項1、請求項2、または請求項3記載のクライアントサーバシステムにおいて、ネットワーク上の全てのアプリケーションサービスを管理するディレクトリサービス手段をネットワーク内に備え、アプリケーションサービスに対する処理の依頼方法に関する情報を前記ディレクトリサービス手段から取得する構成にしたことを特徴とするクライアントサーバシステム。

【請求項5】 請求項1、請求項2、または請求項3記載のクライアントサーバシステムにおいて、アプリケーションサービスに対する処理の依頼方法に関する情報を各アプリケーションサービス手段が管理し、各アプリケーションサービス間で共通な特定のコマンドを各アプリケーションサービス手段へ送ることにより前記依頼方法に関する情報を取得する構成にしたことを特徴とするクライアントサーバシステム。

【請求項6】 請求項1乃至請求項5のいずれかに記載

のクライアントサーバシステムにおいて、クライアント端末装置上で、送られてきた入力項目を含むHTTPデータをブラウザ上に表示し、その入力項目に入力値を入力させるか入力値を選択させるかして指定させ、その入力値をクライアント端末装置から中継サーバへ送り、アプリケーションサービスに対する処理の依頼方法に関する情報をしたがって中継サーバが前記入力値からサービスに対する実際のコマンドおよびパラメータを生成し、アプリケーションサービス手段へ送信することを特徴とするクライアントサーバシステム。

【請求項7】 請求項6記載のクライアントサーバシステムにおいて、中継サーバが、コマンド送信に対する応答としてアプリケーションサービス手段から処理結果を受信し、その処理結果を元にHTTPデータを生成し、生成したHTTPデータをクライアント端末装置へ返送することを特徴とするクライアントサーバシステム。

【請求項8】 請求項7記載のクライアントサーバシステムにおいて、アプリケーションサービス手段へのコマンド送信に対する処理結果としてアプリケーションサービス手段から当該アプリケーションサービスまたは別のアプリケーションサービスのサービス識別子が選択され、中継サーバが選択されたサービス識別子に対応するアプリケーションサービスに端する処理の依頼方法を取得し、取得した情報を基づきHTTPデータを生成し、生成したHTTPデータをクライアント端末装置へ返送することを特徴とするクライアントサーバシステム。

【請求項9】 請求項1、請求項2、または請求項3記載のクライアントサーバシステムにおいて、アプリケーションサービスを、サービスオブジェクトとして外部インターフェースから参照可能な「ドルウェア」を用いて実装し、アプリケーションサービスに対する処理の依頼方法に関する情報をとして前記サービスオブジェクトの外部インターフェースに端する情報をアプリケーションサービス手段から取得する構成にしたことを特徴とするクライアントサーバシステム。

【請求項10】 請求項9記載のクライアントサーバシステムにおいて、前記「ドルウェア」として「AVA(登録商標)」言語のXML/HTTP機能を用い、JAVA言語のselect機能を用いてサービスオブジェクトの外部インターフェースに端する情報を取得することを特徴とするクライアントサーバシステム。

## 【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】本発明は、ネットワークを介してアプリケーションサービスを行なうことができるクライアントサーバシステムに関する。

## 【0002】

【従来の技術】一般的に言って、クライアントサーバシステムとは、所定のアプリケーション処理を行うネットワーク上のアプリケーションサービス装置などアプリケ

3

ーションサービス手数に対してクライアント端末装置が処理を依頼し、その結果を取得するシステムである。このようなシステムでは、アプリケーションサービス手段に対するアクセスを行い、また、アプリケーションサービス手段からの処理結果を受信解析するためのライブラリがクライアント端末装置間に必要で、通常のこのようないライブラリはアプリケーションサービス後に用意される。一般には複数のアプリケーションサービスに対してクライアント端末装置がアクセスする環境ではクライアント端末装置とアプリケーションサービス手段上のプログラム間でのデータ交換プロトコルを統一するために分散型ミドルウェアが用いられることが多い。このようなミドルウェアの例としてはIBMのOS/390、Netscaler社のiQ/DINE、OCNのNSPなどが挙げられる。しかし、通常のこれらのミドルウェアはデータ交換のプロトコルは統一するが、アプリケーションサービスの提供する機能内容、その機能の実現方法、実現結果として選択されるデータの解析方法等に関する情報をクライアント端末装置側が取得することを可能としていない。このため、クライアント端末装置側ではやはりアプリケーションサービスの提供する機能毎に設計されたライブラリを用意し、それをクライアント端末装置間に導入する必要がある。

【0003】ところで、近年の情報技術の普及はほぼ全てのパソコン・ノートパソコン・スマートフォン等を複数させた。そこで、この複数端末を接続したクライアント端末装置とアプリケーションサービス手段との間にHTTPサーバを配置し、HTTPサーバでクライアント端末装置からの指示を受信し、それに基づいてHTTPサーバがネットワーク上のアプリケーションサービス手段に対して処理を依頼し、その処理結果をHTTPデータに変換してHTTPサーバに送信するという複数端末をもつたシステムが多く構築されている。このようなシステムでは、クライアント端末装置上にはHTTPサーバのみが必要で利用者はミドルウェアなど特殊なライブラリをクライアント端末装置に導入する必要から開放される。しかし、このようなシステムでも、ネットワーク上のアプリケーションサービス手段の提供する機能およびそこのアクセス方法はアプリケーションサービス毎に異なるため、HTTPサーバがネットワーク上にあるアプリケーションサービス手段へアクセスし、その結果をHTTPデータに変換するという処理のために繰り返しHTTPサーバに依存したライブラリがHTTPサーバ上に導入されていることが必要になる。そのため、ネットワーク上に新しいアプリケーションサービスが追加された場合には、そのアプリケーションサービスを利用するための処理をHTTPサーバに追加する必要がある。

【0004】

【発明が解決しようとする課題】しかしながら、通常、HTTPサーバ上に表示される内容とネットワーク上のアプリケーションサービス手段が提供する機能との間の一

定の接続規則がないため、従来技術においては、新しいアプリケーションサービスが導入されたときに、それをHTTPサーバから利用するためのHTTPデータのデザインを決定し、HTTPデータとアプリケーションサービスの提供する機能との間での変換方法をアプリケーションサービス毎に開発する必要があった。また、ネットワーク上に各種アプリケーションサービスが一時的に追加されたり、切替されたりするような環境には対応ができないかった。

【0005】本発明の目的は、このような従来技術の問題を解決することにあり、具体的には、クライアントサービス手段において、クライアント端末装置とアプリケーションサービス手段との間に中継サーバを配置し、この中継サーバがクライアント端末装置からの要求に応じてネットワーク上のアプリケーションサービス手段にアクセスする場合に、ネットワーク上に新たなアプリケーションサービスが追加されたときに、中継サーバに新たなモジュールを追加することなく、クライアント端末装置からこの新しいアプリケーションサービスを利用するすることを可能とすることにある。また、アプリケーションサービス手段への処理依頼がアプリケーションサービス手段へのコマンドを送信する形で実行可能なシステムにおいて、アプリケーションサービス手段の各コマンドを利用者が直接指定して実行することが可能なユーティリティーフェームを実現することにある。また、アプリケーションサービス手段への処理依頼がコマンドに加え複数のパラメータを送信する形で実行可能なシステムにおいて、コマンドに付し必要なパラメータを利用者が直接指定して実行することが可能なユーティリティースキームを実現することにある。また、アプリケーションサービス手段への処理依頼がアプリケーションサービス手段へのコマンドを送信する形で実行可能なシステムにおいて、アプリケーションサービスの実行可能なコマンドの一覧を中継サーバに予め登録しておくなくても、新たなサービスが追加された時点で、中継サーバがそれらの情報を動的に収集可能なシステムを実現することにある。また、アプリケーションサービス手段による処理結果がそれと連携する別のアプリケーションサービスの処理と関連する場合、この別のアプリケーションサービス手段へのアクセスを利用して利用者に提供することにある。また、中継サーバが新たなアプリケーションサービスが追加された時点でそれらの情報を動的に収集可能にすることにある。

【0006】

【課題を解決するための手段】上述の課題を解決するため、請求項1記載の発明では、HTTPデータを接続したクライアント端末装置と、そのクライアント端末装置からの要求により処理を行う一つ以上のアプリケーションサービス手段と、HTTPサーバを含む中継サーバとがネットワークを介して接続され、その中継サーバがクライアント端末装置からの要求情報を受信し、その要求情報を基

ついてアプリケーションサービス手段に処理を依頼し、処理結果をそのアプリケーションサービス手段から受信してクライアント端末装置へ返送するクライアントサーバシステムにおいて、中継サーバが、クライアント端末装置からの要求情報中に指定されたアプリケーションサービスについてサービスに対する処理依頼方法に関する情報を取得し、取得した情報を基づいてXMLデータを生成し、クライアント端末装置へ返送することを特徴とする。また、請求項2 記載の発明では、請求項1 記載の発明において、クライアント端末装置で前記要求情報を生成するために、中継サーバが生成するXMLデータが各アプリケーションサービスの各コマンドに対応した入力項目を備え、アプリケーションサービスがクライアント端末装置からの要求により複数の異なる処理を実行可能で、クライアント端末装置からの処理要求が処理内容に対応したコマンドを選択する形で行われる場合、前記入力項目を用いてクライアント端末装置のブラウザ上で入力または選択された内容を中継サーバへ送ることを特徴とする。また、請求項3 記載の発明では、請求項2 記載の発明において、アプリケーションサービス手段が各コマンドの実行にコマンドの指定の他にパラメータの指定が必要な場合、中継サーバが、前記ブラウザ上におけるパラメータの指定を可能にさせる入力項目をもつXMLデータを生成する構造にした。また、請求項4 記載の発明では、請求項1、請求項2、または請求項3 記載の発明において、ネットワーク上の全てのアプリケーションサービスを管理するディレクトリサービス手段をネットワーク内に備え、アプリケーションサービスに対する処理の依頼方法に関する情報を前記ディレクトリサービス手段から取得することを特徴とする。

【 0007 】また、請求項5 記載の発明では、請求項1、請求項2、または請求項3 記載の発明において、アプリケーションサービスに対する処理の依頼方法に関する情報を各アプリケーションサービス手段が管理し、各アプリケーションサービス間で共通な特定のコマンドを各アプリケーションサービス手段へ送ることにより前記依頼方法に関する情報を取得することを特徴とする。また、請求項6 記載の発明では、請求項1 乃至請求項5 のいずれかに記載の発明において、クライアント端末装置上で、送られてきた入力項目を含むXMLデータをブラウザ上に表示し、その入力項目に入力を入力させるか入力欄を選択させかして指定させ、その入力欄をクライアント端末装置から中継サーバへ送り、アプリケーションサービスに対する処理の依頼方法に関する情報をしたがって中継サーバが前記入力欄からサービスに対する実際のコマンドおよびパラメータを生成し、アプリケーションサービス手段へ送信することを特徴とする。また、請求項7 記載の発明では、請求項6 記載の発明において、中継サーバが、コマンド送信に対する応答としてアプリケーションサービス手段から処理結果を受信し、そ

の処理結果を元にXMLデータを生成し、生成したXMLデータをクライアント端末装置へ返送することを特徴とする。また、請求項8 記載の発明では、請求項7 記載の発明において、アプリケーションサービス手段へのコマンド送信に対する処理結果としてアプリケーションサービス手段から当該アプリケーションサービスまたは別のアプリケーションサービスのサービス識別子が返送され、中継サーバが返送されたサービス識別子に対するアプリケーションサービスに対する処理の依頼方法を取得し、取得した情報を基づきXMLデータを生成し、生成したXMLデータをクライアント端末装置へ返送することを特徴とする。また、請求項9 記載の発明では、請求項1、請求項2、または請求項3 記載の発明において、アプリケーションサービスを、サービスオブジェクトとして外部インターフェースから参照可能なミドルウェアを用いて実現し、アプリケーションサービスに対する処理の依頼方法に関する情報をとして前記サービスオブジェクトの外部インターフェースに関する情報をアプリケーションサービス手段から取得することを特徴とする。また、請求項10 記載の発明では、請求項9 記載の発明において、前記ミドルウェアとして並列言語のXML/REFLECTION機能を用い、並列言語のREFLECTION機能を用いてサービスオブジェクトの外部インターフェースに関する情報を取得することを特徴とする。

## 【 0008 】

【 発明の実施の形態】以下、図面により本発明の実施の形態を詳細する。図1は本発明の一実施例を示すクライアントサーバシステムのシステム構成である。図示したように、この実施例のクライアントサーバシステムは、Webブラウザを搭載した1台以上のクライアント端末装置1、それぞれ一つまたは複数のアプリケーションサービス手段を備えた一つ以上のアプリケーションサービス装置2、中継サーバ3、ディレクトリサービス装置4などから構成される。前記において、Webブラウザを搭載したクライアント端末装置1と中継サーバ3との間の通信はHTTPを用いて行われる。このため、中継サーバ3にはHTTPサーバが搭載される。また、各アプリケーションサービス手段と中継サーバ3の間は分散オブジェクト技術を実現するミドルウェアで接続される。このようなミドルウェアの例としてはCORBA、COM Microsoft ActiveX/OLE,DB、Single Object Access Protocol(SOAP)などがあげられる。このミドルウェア機能として、中継サーバ3にはアプリケーションアクセスプロトコルクライアントもしくは、アプリケーションサービス側にはアプリケーションサービスアクセスプロトコルサーバがそれぞれ搭載される。各アプリケーションサービス手段は外部からの処理要求をネットワーク経由で受信し、処理要求に従って処理を実行し、その処理結果を要求元に返送する。各アプリケーションサービス手段への処理指示は処理種類を指定するコマンドとその処理に対するパラメ

7

ータの形で構成される。アプリケーションサービス手段内部では受信したコマンドの種類に応じてコマンドディスパッチャ(*Dispatcher*)により異なるアプリ処理手続きが起動される。また、クライアント端末装置1には、アプリケーションサービスの機構するコマンド一覧およびそのコマンドに対するパラメータの情報を取得する手段を備える。このような手段の実現方法としては、該当するアプリケーションサービス側へ挿入されたコマンドを送信することにより、アプリケーションサービスが認識されているインターフェース(1/4)社機記述データを取得する方法、アプリケーションサービス側が内部の各アプリ処理手続きの持つ属性情報をもつて機記述データを動的に生成して送る方法がある。さらに、ネットワーク上に存在するディレクトリサービス装置4にアプリサービス側の1/4社機記述データを登録しておく方法も可能である。また、Sun Microsystems社の*RPC/JNDI*などのように自己参照機能(*reflection*)を提供するモルウェアの場合、クライアント側モジュールである*proxy*から対応するサーバの属性情報を取得することも可能である。

【0009】ディレクトリサービス装置4はネットワーク上で提供されているアプリケーションサービスの一覧情報を保存している。ディレクトリサービスの実装例としては*slight weight Directory Protocol*(SLDP)、*Service Location Protocol*(SLP)や、Sun Microsystems社の*NCP/registry*機能が挙げられる。*SLP*や*UNI*のように各アプリケーションサービスが自動的にその情報を登録する機能をもつ場合、名前、識別子、ネットワークアドレス等のサービスに該当する基本情報からなるアプリサービス情報をアプリケーションサービスの*Directory Service Agent*が自動的にディレクトリサービス装置4に登録する。*JNDI*などのように自動的な情報の登録機能を持たないディレクトリサービスでは管理者が各アプリケーションサービスのアプリサービス情報を手動で登録する。ディレクトリサービスへのアクセスはそのクライアント機能であるディレクトリサービスクライアント39を介して行われる。ネットワーク上にディレクトリサービスが存在しない構成では、クライアント端末装置上のディレクトリサービスクライアント機能が開通端末(*multicast*または*broadcast*)を利用してネットワーク上の全アプリケーションサービス手段の*Directory Service Agent*にアクセスし、各*Directory Service Agent*が所属するアプリケーションサービスのアプリサービス情報を収容するとシラ構成が可能である。このような実装を可能にするものとしては前述の*Service Location Protocol*(SLP)の他に、Microsoft社の*Universal Plug-and-Play*のためがある。中継サーバ3は*WEB*ブラウザを搭載したクライアント端末装置1からの要求を*HTTP*で受信し、ネットワーク上のアプリケーションサービスの一覧とコマンド一覧を*HTML*データで送信する。さらに、クライアント

8

端末装置1からの要求により特定のアプリケーションサービス手段へコマンドを送信し、そのコマンド実行結果をそのアプリケーションサービス手段から受信し、クライアント端末装置1へ送信する。

【0010】図1に示したように、中継サーバ3には、*HTTP*サーバ(1/4)を介して受信したクライアント端末装置1からの要求を処理するクライアント制御部32、ディレクトリサービス装置4へのアクセスその他の方法でネットワーク上のアプリケーションサービスの一覧を取得して*HTML*データを生成させるサービス一覧取得部33、特定のアプリケーションサービスの1/4社機記述データを取得して*HTML*データとして生成する1/4社機数得部34などがある。さらに、クライアント制御部32により起動され、アプリケーションサービスに対するコマンドを生成し、それをアプリサービスアクセスプロトコル(クライアント)35を介して実際のアプリケーションサービス手段へ送信し、次にそのコマンドの処理結果(実行結果)を受信し、その処理結果を実行結果解析部36を通して*HTML*データを生成し、それをクライアント制御部32へ送信するサービス起動制御部37がある。また、実際に*HTML*データを生成する手段として、サービス一覧取得部33から起動されるサービス一覧生成部、1/4社機数得部34から起動されるコマンド一覧生成部とコマンド引数入力項目生成部、実行結果解析部36から起動されるコマンド実行結果生成部などから成る*HTML*生成部38がある。

【0011】図9に、この実施例の処理の流れを示す。処理はクライアント端末装置1内の*WEB*ブラウザが中継サーバ3にサービス一覧を要求することから開始する。*WEB*ブラウザからの要求は中継サーバ3の1/4社機記述部1で受信され、クライアント制御部32に通知される。クライアント制御部32はサービス一覧取得部33を呼び出す。サービス一覧取得部33はディレクトリサービスクライアント39を利用してネットワーク上のディレクトリサービス装置4にアクセスし、ネットワーク上のアプリケーションサービスの一覧を取得し、その結果を*HTML*生成部38のサービス一覧生成機能を用いて、図3のような*HTML*データを生成する。この*HTML*データは、利用者がいずれかのアプリケーションサービスを選択でき、さらに、選択されたアプリケーションサービスの識別子を中継サーバ3へ再度通知できるように構成されている。例えば、この*HTML*データに記述される情報はアプリケーションサービスの名前と識別子であり、他にディレクトリサービスから取得できるアプリサービス情報として組織や機能情報など利用者に提供されるべき情報などを含む。図2に示した例ではアプリケーションサービスの識別子としてアプリケーションサービスの*URI*を用いているが、中継サーバ3とアプリケーションサービス間のモルウェアにより、アプリケーションサービスを特定できるものならばURIなしにかぎらない識別子が

9

利用可能である。また、この例では利⽤者の入力手段としてアプリケーションサービス毎のハイパーテインクを利用していいるが、同じくCOMのCOM機能を使い、アプリケーションサービス名を利⽤者がリストボックス等から選択し、送信ボタンを押すことで選択されたアプリケーションサービスの識別子を中継サーバ3へ送るようにCOMを構成することも可能である。生成されたCOMデータはクライアント端末装置3からHTTPサーバ3へ介して要求元のWWWブラウザに送信される。なお、この実施例では、ネットワーク上のアプリケーションサービスの一覧を保持したディレクトリサービス装置4がネットワーク上に存在する構成を示しているが、前述のように、ディレクトリサービス装置4がない構成でも、ディレクトリサービスクライアント端末がネットワーク上の各アプリケーションサービスのDirectory Service Agentに直接アクセスする方法や、各アプリケーションサービスのDirectory Service Agentに直接アクセスする構成も可能である。

[ 0 0 1 2 ] クライアント端末装置1のWWWブラウザでは、このCOMデータを受信すると、例えば図3に示したように表示する。次に、この表示されたCOMデータから利⽤者が特定のアプリケーションサービス名を選択すると、選択されたアプリケーションサービスの識別子を含む、選択された情報がCOMデータで再び中継サーバ3へ送られ、クライアント端末装置3に通知される。クライアント端末装置3では受信した情報をから選択されたアプリケーションサービスの識別子を抽出し、その結果を1/P仕様取得部3-4へ送り、該当するアプリケーションサービスの1/P仕様取得データに対応したCOMデータを生成する。さらに、1/P仕様取得部3-4では、前述のように、識別子で指定されたアプリケーションサービスへサービス端で実行のコマンドを送信するか、ディレクトリサービス装置4に記録されている情報を取得するか、クライアント側モジュールであるproxymから対応するアプリケーションサービスの1/P仕様を取得するかして、1/P仕様取得データを取得する。即ち、この際に得られる仕様の記述方法がCOMデータ以外で記述されており、1/P仕様取得部3-4が一定のルールに基づいてCOMデータに変換することが本発明の主題である。1/P仕様取得はCOMデータへの変換アルゴリズムが定義できるものであればどのような形式でも構わない。このようない1/P仕様取得の例として図4にCOMのCOM記述の例を示す。あるCOMのCOM記述のCOM言語では、各アプリケーションサービスのプログラムオブジェクトに、それ自身の持つ外部1/P関数の仕様を実行時に問い合わせることが可能で、アプリケーションサービスがCOM言語で表現されている場合、アプリケーションサービス手段内に内蔵する各プログラムオブジェクトの外部1/P関数仕様をクライアント端末装置1からの要求時に収集し、動的に例えばCOM形式の1/P仕様取得を生成することが可能

10

である。また、COM言語をベースとしたCOM/GUIはドルウェア環境では、中継サーバ内でクライアント側モジュールであるproxymから対応するアプリケーションサービスの1/P仕様をCOMのプログラムオブジェクトとして取得することが可能である。図5に、この場合のCOM言語でのコード例を示す。この例では1/P仕様記述を独自の形式で文字列データとして生成している。1/P仕様取得部3-4は1/P仕様記述を取得後、それをCOM生成部3-8のコマンド一覧生成部は1/P仕様記述をもコマンドを抽出し、各コマンド毎にコマンド引数入力項目生成部を呼び出してコマンド名とパラメータに対応する<COM>/<COM>項目を生成する。COM/GUI、COMなどもドルウェアがリモート中継を呼び出しの形式を実現しても場合、コマンドは各外添1/P関数名、パラメータは各関数の引数が採用される。

[ 0 0 1 3 ] この場合の<COM>の構成例を図6に示す。ここでの構成例では、各パラメータに対応する<COM>タグ、および送信ボタンに割り当てる<INPUT type="button" name="command">を配置する。コマンド名は<COM type="button" name="command">の属性既然としてセクトする。また、<COM>タグの属性として送信ボタンが押下されたときに、コマンド名およびパラメータ値が送信される送信先URLとして中継サーバ3のHTTPサーバ3-1のURLを指定する。コマンド一覧生成部はコマンド毎にコマンド引数入力項目生成部を呼び出してこれらの各コマンドに対応する<COM>タグを生成し、1/P仕様取得部3-4はこれをクライアント端末装置1のWWWブラウザに送信される。図7に生成されるCOMデータの例とWWWブラウザ上での表示例を示す。WWWブラウザ上で利⽤者が一つのコマンドに対し各パラメータを⼊力してコマンド名でラベルされたボタンを押下すると、クライアント端末装置1は図8に示すようなCOMメッセージを送信する。中継サーバ3のHTTPサーバ3-1はこれを受け取り、アプリサーバ端末装置、コマンド名、パラメータ値をセットでクライアント端末装置3-2へ送信する。クライアント端末装置3-2ではこれをサービス起動制御部3-7へ送信する。サービス起動制御部3-7では、コマンド生成手段を通過してアプリケーションサービスにアクセスするためのドルウェアに対応した形で図8に示す変換を行い、コマンドを生成し、アプリサーバ端末装置で選択されるアプリケーションサービス宛てに送信する。これにより、そのアプリケーションサービスからコマンド実行結果(処理結果)が戻ってくると、サービス起動制御部3-7はその結果をコマンド実行結果解析部3-6に渡し、その通り値の型に応じたCOMデータを生成し、そのCOMデータをクライアント端末装置3-2および中継サーバ3-1を経由して当該クライアント端末装置1へ送り、WWWブラウザに表示する。この際、コ

II

マンド 実行結果がアプリケーションサービス識別子であった場合には識別子を追跡データとして選択するではなく、この識別子をクライアント端末装置3に渡し、クライアント端末装置3では、1つ社様取得部3.4を起動して識別子に対するアプリケーションサービスの1つ社様を得し、そのxmlデータを生成し、それをxmlプラウザに送信する。

## 【 0014 】

【発明の効果】以上説明したように、本発明によれば、請求項1記載の発明では、xmlデータを含む中継サーバ10がクライアント端末装置からの要求情報を受信し、その要求情報に基づいてアプリケーションサービス手段に処理を依頼し、処理結果をそのアプリケーションサービス手段から受信してクライアント端末装置へ返送する際、クライアント端末装置からの要求情報を元に指定されたアプリケーションサービスについてサービスに対する処理依頼方法に関する情報が中継サーバにより取得され、取得された情報に基づいてその中継サーバによりxmlデータが生成され、クライアント端末装置へ返送されるもので、ブラウザを利用したクライアントサーバシステムにおいて、ネットワーク上へ新たなアプリケーションサービスが追加された時点で、自動的に、クライアント端末装置からアプリケーションサービスが利用できるようになる。また、請求項2記載の発明では、請求項1記載の発明において、クライアント端末装置からの処理要求が処理内容に対応したコマンドを送信する形で行われる場合、中継サーバも送られてきた入力項目を用いてクライアント端末装置のブラウザ上で入力または選択された内容が中継サーバへ送られるので、アプリケーションサービスの各コマンドを利用者が直接指定して、新しいアプリケーションサービスの提供する全機能をクライアント端末装置から利用することができます。また、請求項3記載の発明では、請求項2記載の発明において、アプリケーションサービス手段が各コマンドの実行にコマンドの指定の他にパラメータの指定が必要な場合、中継サーバにより、ブラウザ上におけるパラメータ値の指定を可能にさせる入力項目をもつxmlデータが生成されるので、パラメータの指定が必要な場合でも、新しいアプリケーションサービスの提供する全機能をクライアント端末装置から利用することができます。また、請求項4記載の発明では、請求項1、請求項2、または請求項3記載の発明において、ネットワーク上の全てのアプリケーションサービスに対する処理の依頼方法に関する情報がデータクトリーサービス手段から取得されるので、その依頼方法を中継サーバに手め渡しておかないとも、新たなサービスが追加された時点で、中継サーバがそれらの情報を自動的に収集できる。

【 0015 】また、請求項5記載の発明では、請求項1、請求項2、または請求項3記載の発明において、アプリケーションサービスに対する処理の依頼方法に関する

III

る情報が各アプリケーションサービス手段により管理され、各アプリケーションサービス間で共通な特定のコマンドを各アプリケーションサービス手段へ送ることにより前述依頼方法に関する情報が取扱われる所以、同時に、その依頼方法を中継サーバに予め登録してもかなくとも、新たなサービスが追加された時点で、中継サーバがそれらの情報を自動的に収集できる。また、請求項6記載の発明において、クライアント端末装置上で、送られてきた入力項目をもつxmlデータがブラウザ上に表示され、利用者がその入力項目に入力値を入力するか入力値を選択するかすることにより要求内容がクライアント端末装置から中継サーバへ送られ、アプリケーションサービスに対する処理の依頼方法に関する情報にしたがって、中継サーバにより前記入力値からサービスに対する実際のコマンドおよびパラメータが生成され、アプリケーションサービス手段へ送信されるので、簡単な操作で所要のアプリケーションサービス内容を指定してそのアプリケーションサービス手段を実行させることができます。また、請求項7記載の発明では、請求項8記載の発明において、中継サーバにより、コマンド送信に対する応答としてアプリケーションサービス手段から処理結果が受けられ、その処理結果を元にxmlデータが生成され、生成されたxmlデータがクライアント端末装置へ返送されるので、簡単な操作で所要のアプリケーションサービス内容を指定してそのアプリケーションサービスの結果をクライアント端末装置で得ることができる。また、請求項8記載の発明では、請求項7記載の発明において、アプリケーションサービス手段へのコマンド送信に対する処理結果としてアプリケーションサービス手段から当該アプリケーションサービスまたは別のアプリケーションサービスのサービス識別子が返送され、中継サーバにより、返送されたサービス識別子に対するアプリケーションサービスに対する処理の依頼方法が取扱され、取扱された情報に基づきxmlデータが生成され、生成されたxmlデータがクライアント端末装置へ返送されるので、アプリケーションサービス手段による処理結果がそれと連携する別のアプリケーションサービスの処理と関連する場合、容易にこの別のアプリケーションサービスも受けられる。

【 0016 】また、請求項9記載の発明では、請求項1、請求項2、または請求項3記載の発明において、アプリケーションサービスが、サービスオブジェクトとして外部インターフェースから参照可能なドキュメントを用いて表記され、アプリケーションサービスに対する処理の依頼方法に関する情報としてそのサービスオブジェクトの外部インターフェースに該する情報がアプリケーションサービス手段から取扱されるので、アプリケーションサービスがドキュメントを用いてサービスオブジェクトとして実装されている場合においても、請求項1、

13

請求項2、または請求項3 記載の発明の効果を実現することができる。また、請求項1.0 記載の要件では、請求項3 記載の説明において、モデルウェアとしてXML語彙のみ/CGI機能が用いられ、XML語彙のnavigation機能を用いてサービスオブジェクトの外部インターフェースに関する情報が取得されるので、アプリケーションサービスがXML語彙のnavigation機能を用いてサービスオブジェクトとして実装されている場合においても、請求項1、請求項2、または請求項3 記載の発明の効果を実現することができる。

【 図面の簡単な説明】

【 図1】本発明の一実施例を示すクライアントサーバンシステムのシステム構成図である。

【 図2】本発明の一実施例を示すクライアントサーバンシステムの説明図である。

【 図3】本発明の一実施例を示すクライアントサーバンシステムの他の説明図である。

【 図4】本発明の一実施例を示すクライアントサーバンシステムの他の説明図である。

【 図5】本発明の一実施例を示すクライアントサーバンシステムの他の説明図である。

14

【 図6】本発明の一実施例を示すクライアントサーバンシステムの他の説明図である。

【 図7】本発明の一実施例を示すクライアントサーバンシステムの他の説明図である。

【 図8】本発明の一実施例を示すクライアントサーバンシステムの他の説明図である。

【 図9】本発明の一実施例を示すクライアントサーバンシステムのフロー図である。

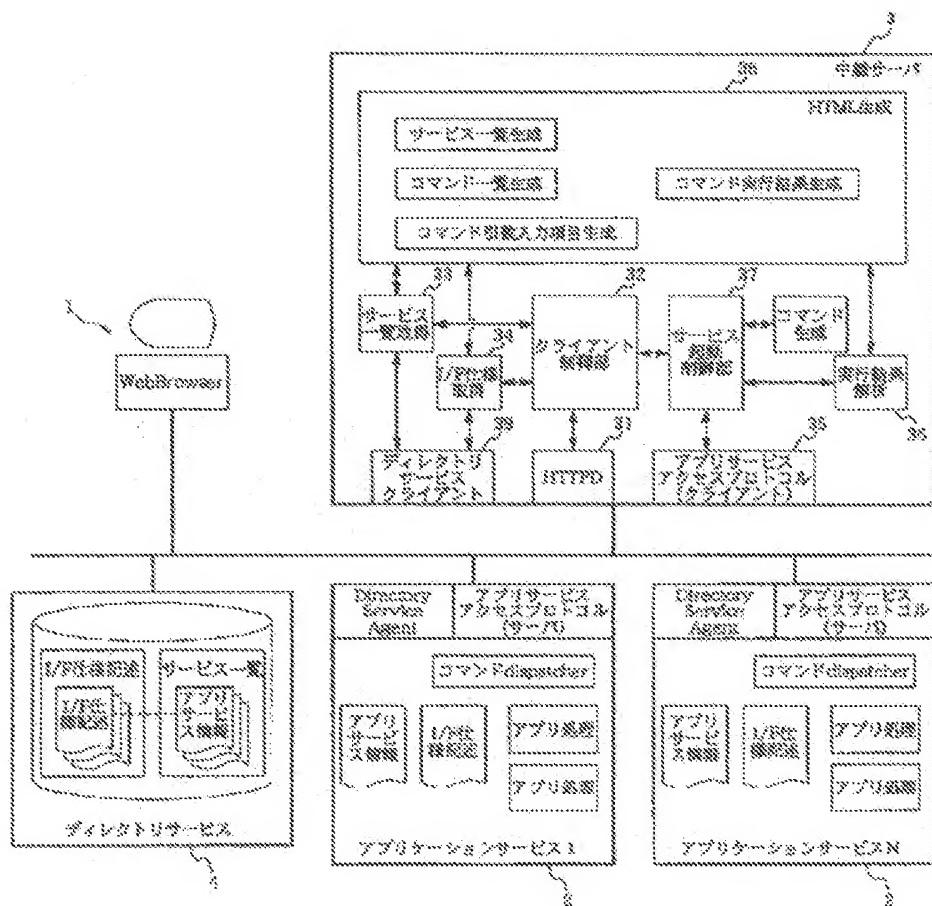
【 付号の説明】

- 1 クライアント端末装置
- 2 アプリケーションサービス装置
- 3 中継サーバ
- 4 ディレクトリサービス装置
- 5.1 HTTPサーバ
- 5.2 クライアント制御部
- 5.3 サービス一覧取得部
- 5.4 ファイル取得部
- 5.5 アプリサービスプロトコル部
- 5.6 実行結果解析部
- 5.7 サービス起動制御部
- 5.8 XML生成部

### 【 図2】

```
<html>
<head>
<title>
</title>
<body>
<a href="http://getway.xxx.co.jp/plentry?servID=http%3A%2F%2FprintSRV.xxx.co.jp">EUS 7サービス</a>
<a href="http://getway.xxx.co.jp/plentry?servID=http%3A%2F%2FdisplaySRV/xxx.co.jp">EUS 8サービス</a>
<a href="http://getway.xxx.co.jp/plentry?servID=http%3A%2F%2FaxSRV.xxx.co.jp">FAX装置サービス</a>
<a href="http://getway.xxx.co.jp/plentry?servID=http%3A%2F%2FWORDPS/xxx.co.jp">WORDPSサービス</a>
<a href="http://getway.xxx.co.jp/plentry?servID=http%3A%2F%2FmailSRV.xxx.co.jp">メールサービス</a>
<a href="http://getway.xxx.co.jp/plentry?servID=http%3A%2F%2FMSWORD/xxx.co.jp">MS-WORDサービス</a>
```

[ 図1 ]

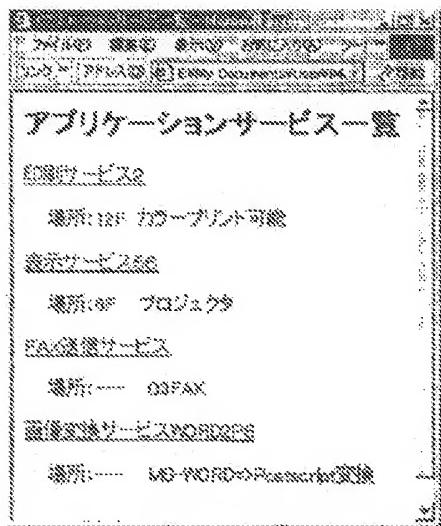


[ 図4 ]

```
module Apiserv1 {
    interface print {
        typedef long jobID;
        typedef string svrID;
        jobID print(in string string_to_print);
        jobID printURL(in string url);
        svrID getControl();
    }

    interface control {
        typedef long jobID;
        jobID currentJob();
        void cancelJob(in jobID jobid);
    }
}
```

【 図3 】



【 図5 】

```

import java.lang.reflect.*;
import java.awt.*;

class SampleClass {

    String getIPInfo(Object proxy) {
        String out;
        Class c = proxy.getClass();
        Method[] methodArray = c.getMethods();
        for (int i = 0; i < methodArray.length; i++) {
            out = "name:" + methodArray[i].getName();
            out = out + " type:" + methodArray[i].getReturnType().getName();
            Class[] parameterTypes = methodArray[i].getParameterTypes();
            for (int k = 0; k < parameterTypes.length; k++) {
                out = out + " para:" + parameterTypes[k].getName();
            }
        }
        return out;
    }
}
    
```

## 【 図6 】

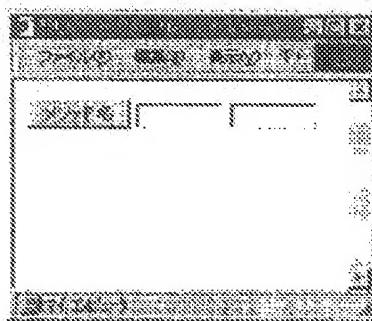
インターフェースシグチャ:

メソッド名 (パラメータ1,...,パラメータN)



出力 HTML :

```
<form method="GET" action="中華サーフのURL">
<input type="submit" value="メソッド名: name=メソッド名">
<input type="text" name="パラメータ1" size="10">
<br>
<input type="text" name="パラメータN" size="10">
</form>
```



## 【 図8 】

入力 HTTP:

GET /commands/メソッド名&amp;#3D;アリサー/アドレス&amp;#3D;第一引数&amp;#3D;第二引数 HTTP/1.1



コマンド実行:

アリサー/アドレスで指定されるアリダーションサーポスに対して次のコマンドを送信する。

メソッド名 (第一引数の値, 第二引数の値, ... 第N引数の値)

[ 図7 ]

```

<html>
<body>
<h2>印刷サーバのコマンド</h2>

<form method="GET" action="http://gateway.xxx.co.jp/entry">
    <!--action 属性に中継サーバの URL -->
    <input type="hidden" name="srvID" value="http%3A%2F%2FprintSRV.xxx.co.jp">
        <!--hidden 項目としてアプリサーバの識別子 -->
    <input type="submit" value="print" name="command">
    <input type="text" name="arg1" size="10">
</form>

<form method="GET" action="http://gateway.xxx.co.jp/entry">
<input type="hidden" name="srvID" value="http%3A%2F%2FprintSRV.xxx.co.jp">
<input type="submit" value="printURL" name="command"> </form>

<form method="GET" action="http://gateway.xxx.co.jp/entry">
<input type="hidden" name="srvID" value="http%3A%2F%2FprintSRV.xxx.co.jp">
<input type="submit" value="currentJOB" name="command">
</form>

<form method="GET" action="http://gateway.xxx.co.jp/entry">
<input type="hidden" name="srvID" value="http%3A%2F%2FprintSRV.xxx.co.jp">
<input type="submit" value="cancelJOB" name="command">
<input type="text" name="arg1" size="10">
</form>

</body>
</html>

```

